

SpringAOP面向方面编程原理：AOP概念Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_SpringAOP_E9_c104_644660.htm AOP(Aspect Oriented Programming)，也就是面向方面编程的技术。AOP基于IoC基础，是对OOP的有益补充。AOP将应用系统分为两部分，核心业务逻辑（Core business concerns）及横向的通用逻辑，也就是所谓的方面Crosscutting enterprise concerns，例如，所有大中型应用都要涉及到的持久化管理（Persistent）、事务管理（Transaction Management）、安全管理（Security）、日志管理（Logging）和调试管理（Debugging）等。AOP正在成为软件开发的下一个光环。使用AOP，你可以将处理aspect的代码注入主程序，通常主程序的主要目的并不在于处理这些aspect。AOP可以防止代码混乱。Spring framework是很有前途的AOP技术。作为一种非侵略性的、轻型的AOP framework，你无需使用预编译器或其他元标签，便可以在Java程序中使用它。这意味着开发团队里只需一人要对付AOP framework，其他人还是像往常一样编程。AOP概念让我们从定义一些重要的AOP概念开始。

方面（Aspect）：一个关注点的模块化，这个关注点实现可能另外横切多个对象。事务管理是J2EE应用中一个很好的横切关注点例子。方面用Spring的Advisor或拦截器实现。

连接点（Joinpoint）：程序执行过程中明确的点，如方法的调用或特定的异常被抛出。

通知（Advice）：在特定的连接点，AOP框架执行的动作。各种类型的通知包括“around”、“before”和“throws”通知。通知类型将在下面讨论。许多AOP框架包括Spring都是以拦截器做通知模型，维护一个“

围绕”连接点的拦截器链。 切入点 (Pointcut) : 指定一个通知将被引发的一系列连接点的集合。AOP框架必须允许开发者指定切入点, 例如, 使用正则表达式。 引入

(Introduction) : 添加方法或字段到被通知的类。Spring允许引入新的接口到任何被通知的对象。例如, 你可以使用一个引入使任何对象实现IsModified接口, 来简化缓存。 目标对象

(Target Object) : 包含连接点的对象, 也被称作被通知或被代理对象。 AOP代理 (AOP Proxy) : AOP框架创建的对象, 包含通知。在Spring中, AOP代理可以是JDK动态代理或CGLIB代理。 编织 (Weaving) : 组装方面来创建一个被通知对象。这可以在编译时完成 (例如使用AspectJ编译器) , 也可以在运行时完成。Spring和其他纯Java AOP框架一样, 在运行时完成织入。 各种通知类型包括: 100Test 下载频道 开通, 各类考试题目直接下载。详细请访问 www.100test.com